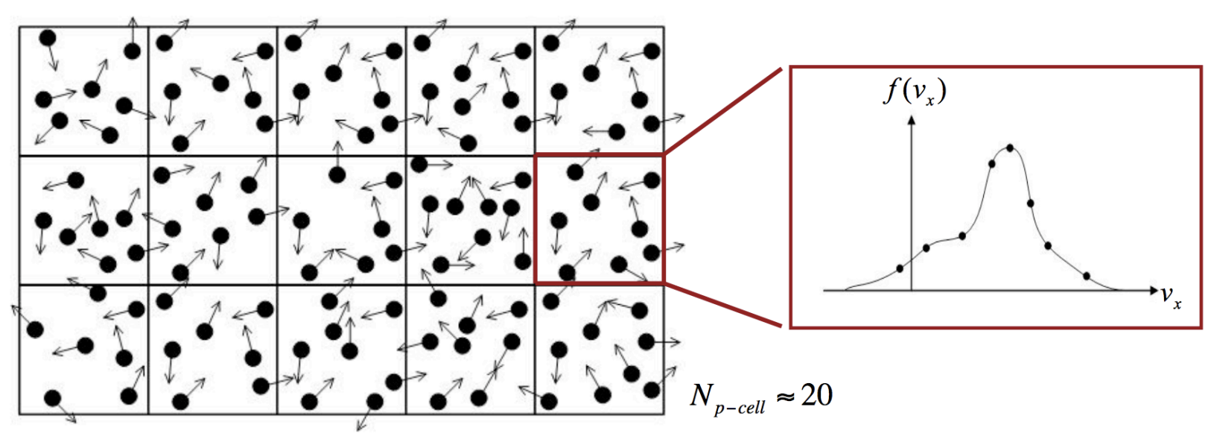
DSMC Solutions of Hypersonic Flow over a Planetary Probe using Automated Adaptive Mesh Refinement and Cut-Cell Algorithms

University of Minnesota
Supercomputing Institute

Thomas E. Schwartzentruber
Department of Aerospace Engineering and Mechanics
The University of Minnesota, Twin Cities Campus, Minneapolis, MN, 55455 USA
schwartz@aem.umn.edu

1 The Direct Simulation Monte Carlo Method



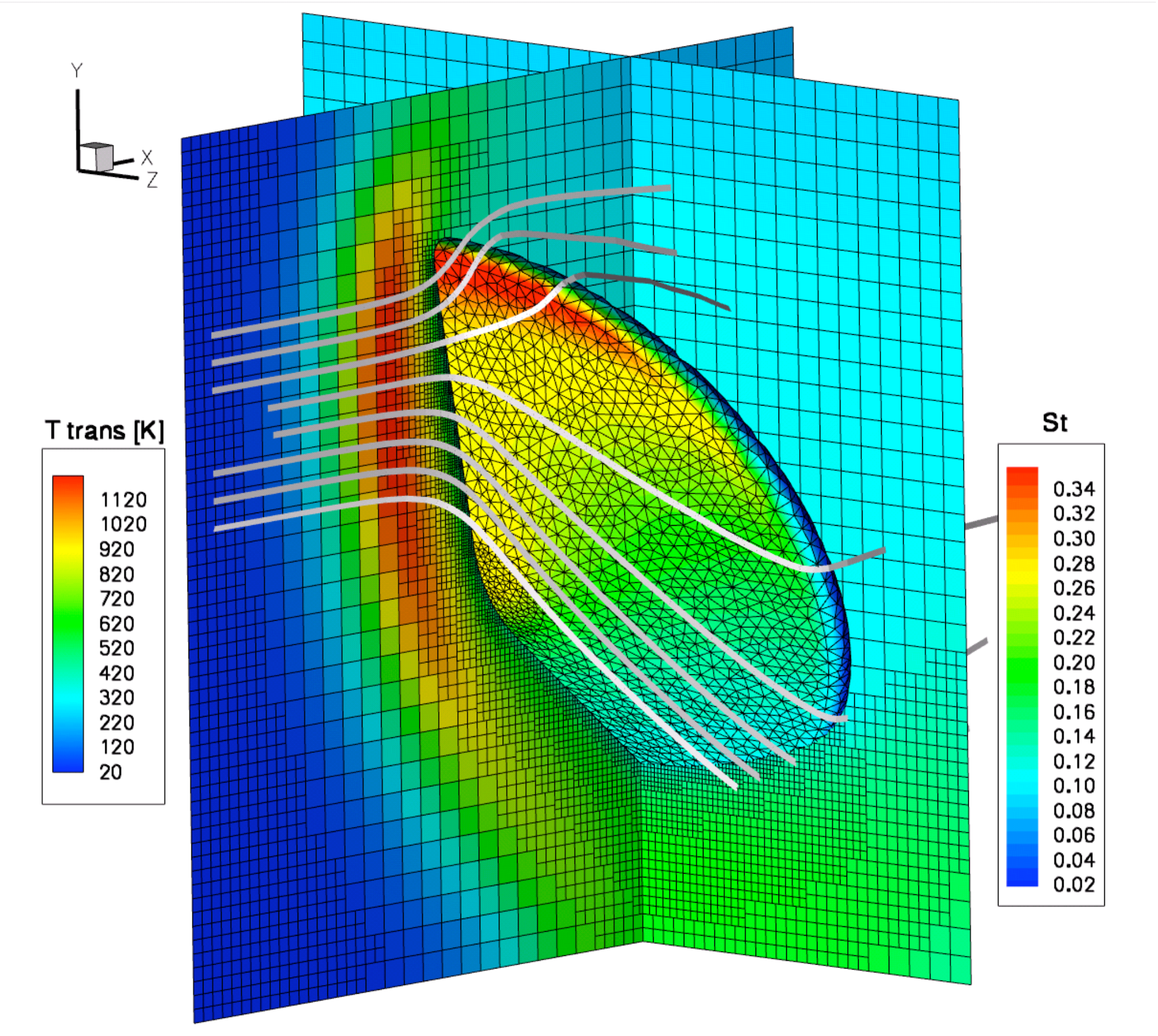
Schematic of the DSMC particle method.

Direct simulation Monte Carlo (DSMC) is a particle method that simulates the Boltzmann equation. The Boltzmann equation is the governing equation for dilute gases ranging from continuum to free-molecular conditions and models the evolution of the velocity distribution function, $f(\mathbf{x}, \mathbf{v}, t)$. The DSMC method is therefore able to accurately model non-equilibrium (low-density, rarefied) flows and is also accurate for continuum flows where it reproduces results from continuum computational fluid dynamics (CFD). The DSMC method tracks a large number of simulated gas molecules through a computational domain allowing for collisions between gas molecules and also collisions with surfaces. The main assumption inherent in the DSMC method is that molecular movement can be decoupled from molecular collisions (an accurate assumption for dilute gases). Locally, as long as:

- the cell size is less than the mean-free-path ($\Delta x < \lambda$) ,
- the time step is less than the mean-collision-time ($\Delta t < \tau_c$),
- there are a sufficient number of simulated particles per cell ($N_p \geq 20$), and
- the collision-rate and reaction-probabilities are chosen correctly,

the DSMC method is able to accurately simulate real, complex gas flows. For example, NASA has used DSMC simulations to support many missions including Space Shuttle missions, Cassini/Huygens, Mars Reconnaissance Orbiter, Mars Global Surveyor, Mars Science Laboratory, Phoenix Mars Lander, Mars Sample Return, and the X-37.

1.1 DSMC Simulation Basics



3D DSMC simulation of flow over a planetary probe at 30 degrees angle of attack using the MGDS code developed at the University of Minnesota. Flow conditions include: [N_2 gas, $M = 20.2$, $\rho = 1.73 \times 10^{-5}\ kg/m^3$, $T = 13.3\ K$].

A typical single-processor DSMC simulation (image right) may contain 100,000 computational cells, with roughly 20-50 particles per cell, resulting in 2-5 million simulation particles. The general algorithm is:

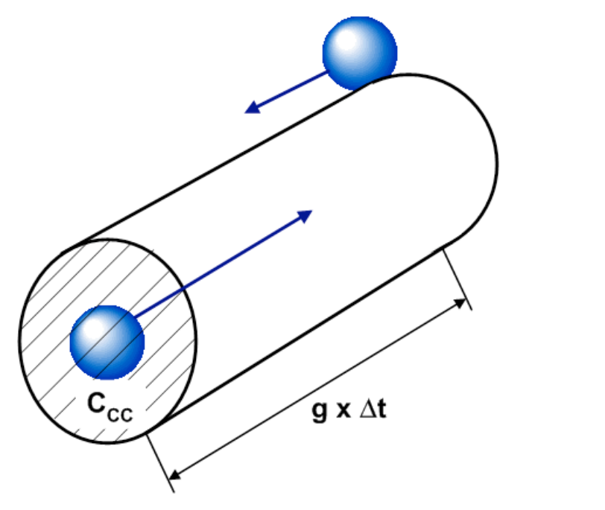
- move all particles according to their molecular velocity for Δt
- generate/move inflow particles, remove outflow particles, compute results of surface collisions
- collide particles within the same cell, update post-collision velocities, and internal energies
- sample macroscopic cell properties, and then repeat

Particles are moved and sorted via "ray tracing". Ray tracing is generic to any type of 3D mesh (Cartesian, hexahedral, tetrahedral, mixed, etc.) and computing particle collisions with arbitrary triangulated surface geometries is naturally handled. Ray tracing from cell to cell may seem unnecessary for Cartesian meshes, however, it is ultimately required in the vicinity of surfaces and when variable time steps are utilized within each cell.

Within a cell, all particle pairs are collided with probability:

$$P_{collision} = \frac{W_p(C_{cc} g \Delta t)}{V_{cell}} \tag{1}$$

where W_p (the particle weight) is the number of real molecules represented by each simulated particle, g is the relative velocity of the two particles, and C_{cc} is the collision cross-section of the interaction.



The collision cross-section can be thought of as the extent of influence of the molecules' inter-atomic potential. The numerator of Eq. 1 represents the volume swept out by the molecules' interatomic potentials during a single timestep (image left). For example, if this volume is equal to the volume of the cell, then the probability of that particle colliding with another in the same cell during that time step is unity. Typically only 10% of DSMC particles collide during each timestep. A simple and accurate model for the collision cross-section is the Variable Hard Sphere (VHS) model, where the hard sphere cross-section, based on a reference molecule diameter (d_{ref}), is modified as a power-law function of the relative collision velocity. This model is physically realistic from the molecular perspective and the resulting collision rate ($\nu = 1/\tau_c$) reproduces real gas viscosities (μ) and thermal conductivities:

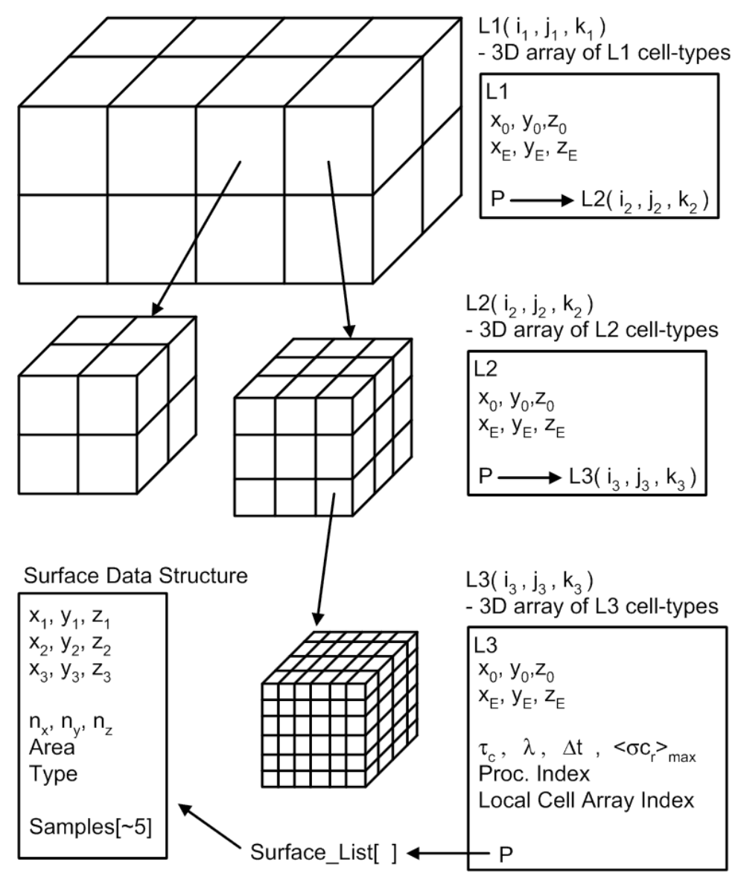
$$C_{cc} = \frac{\pi d_{ref}^2}{g^{2\zeta}} \quad \longrightarrow \quad \mu = \mu_{ref}\left(\frac{T}{T_{ref}}\right)^{\omega}, \quad \text{where} \quad \omega = \frac{1}{2} + \zeta \tag{2}$$

Once a collision is selected, a random scattering angle is typically chosen and an elastic collision is carried out. If internal energy transfer (rotation, vibration) and chemical reactions are important, probabilities of such events are specified in terms of the total collision energy. Parameters such as the rotational and vibrational collision numbers and steric factors for reactions are derived to reproduce existing reaction rate data commonly employed by continuum CFD simulations. Thus the DSMC method and molecular-based collision models are able to accurately model highly non-equilibrium processes while also producing consistent results with CFD in the continuum regime.

2 The Molecular Gas Dynamic Simulator (MGDS) Code

2.1 Geometry Data Structure

The MGDS code employs a separate Geometry data structure (image below) that holds the 3-level Cartesian mesh geometry and triangulated surface geometry. The entire Cartesian mesh requires very little memory storage and large portions of the Geometry data structure (often the entire mesh) can be stored on every partition in a large parallel simulation. Each Level-1 (L1) cell contains any number of L2 cells, each of which contains any number of L3 cells. The Geometry data structure is designed to contain all information necessary for adaptive mesh refinement and particle movement routines.



L1(i_1 , j_1 , k_1)
- 3D array of L1 cell-types

L1
x_0, y_0, z_0
x_E, y_E, z_E
P → L2(i_2 , j_2 , k_2)

L2(i_2 , j_2 , k_2)
- 3D array of L2 cell-types

L2
x_0, y_0, z_0
x_E, y_E, z_E
P → L3(i_3 , j_3 , k_3)

Surface Data Structure
x_1, y_1, z_1
x_2, y_2, z_2
x_3, y_3, z_3
n_x, n_y, n_z
Area
Type
Samples[~5]

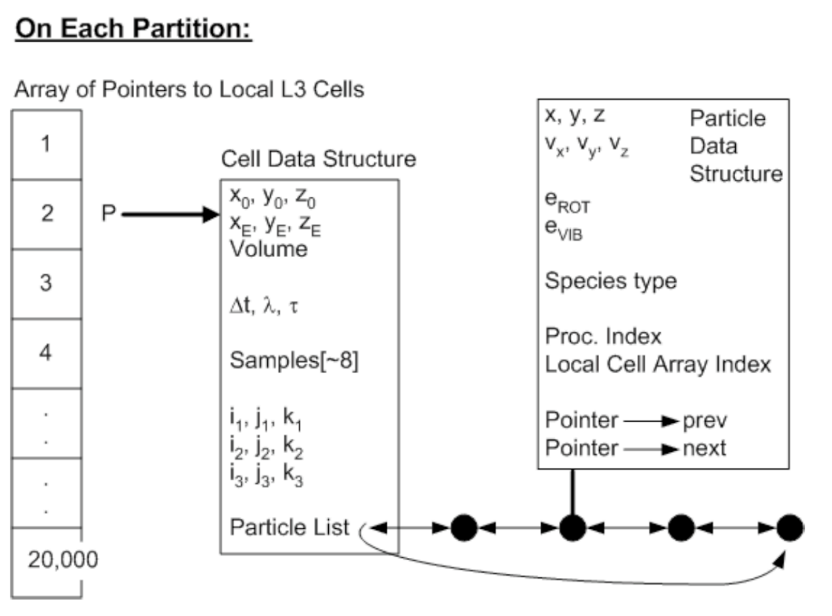
L3(i_3 , j_3 , k_3)
- 3D array of L3 cell-types

L3
x_0, y_0, z_0
x_E, y_E, z_E
τ_c , λ , Δt , $<\sigma c_r>_{max}$
Proc. Index
Local Cell Array Index
Surface_List[ ] ← P

This results in minimal parallel communication when particles move across partitions. Essentially, since each partition "knows the entire mesh", all particles can be moved for a complete time step even if they cross partitions, collide with surfaces, and move with variable time steps.

2.2 Cell/Particle Data Structures

The largest memory requirements are associated with Cell and Particle data structures that must be partitioned. Each cell must store cumulative samples of mass, momentum, and energy (used to determine macroscopic properties) and pointers to the head and tail of a doubly-linked list of Particle data structures. Each Particle structure contains position, velocity, internal energy, and species information.



On Each Partition:

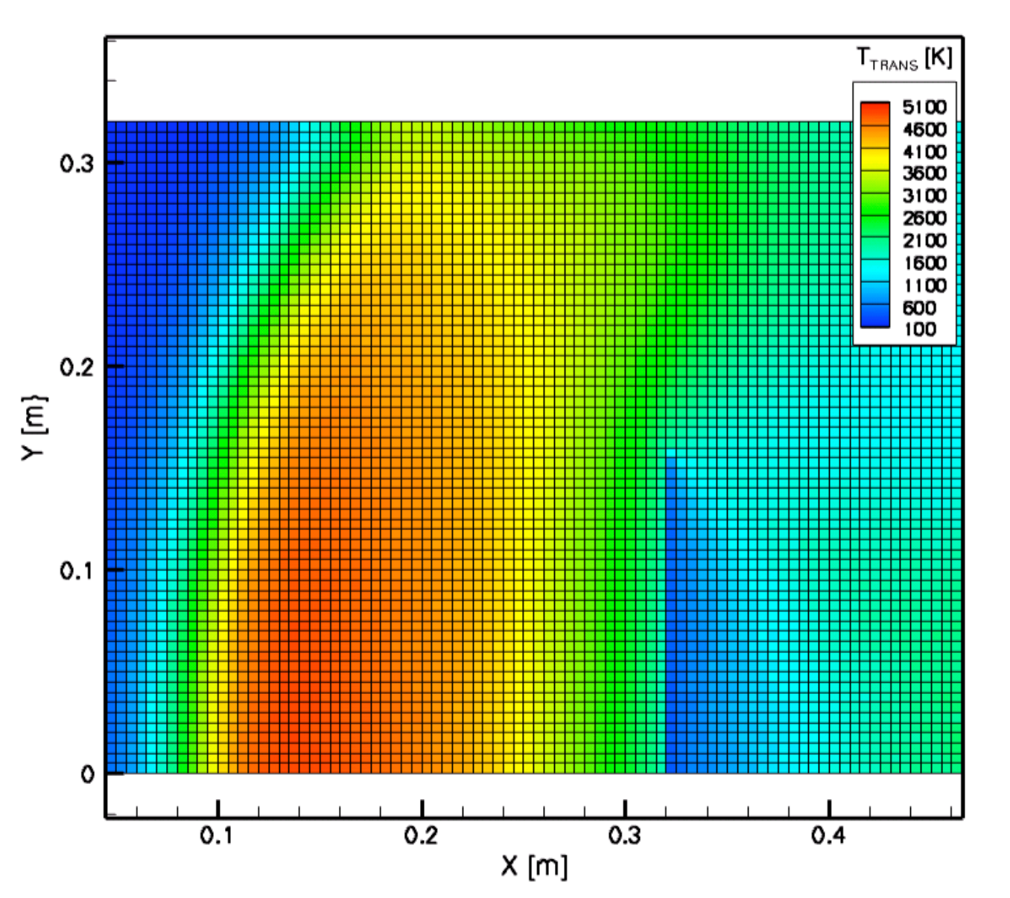
Array of Pointers to Local L3 Cells
1
2
3
4
.
.
20,000

Cell Data Structure
P → x_0, y_0, z_0
x_E, y_E, z_E
Volume
Δt, λ, τ
Samples[~8]
i_1, j_1, k_1
i_2, j_2, k_2
i_3, j_3, k_3
Particle List

Particle Data Structure
x, y, z
v_x, v_y, v_z
e_{ROT}
e_{VIB}
Species type
Proc. Index
Local Cell Array Index
Pointer → prev
Pointer → next

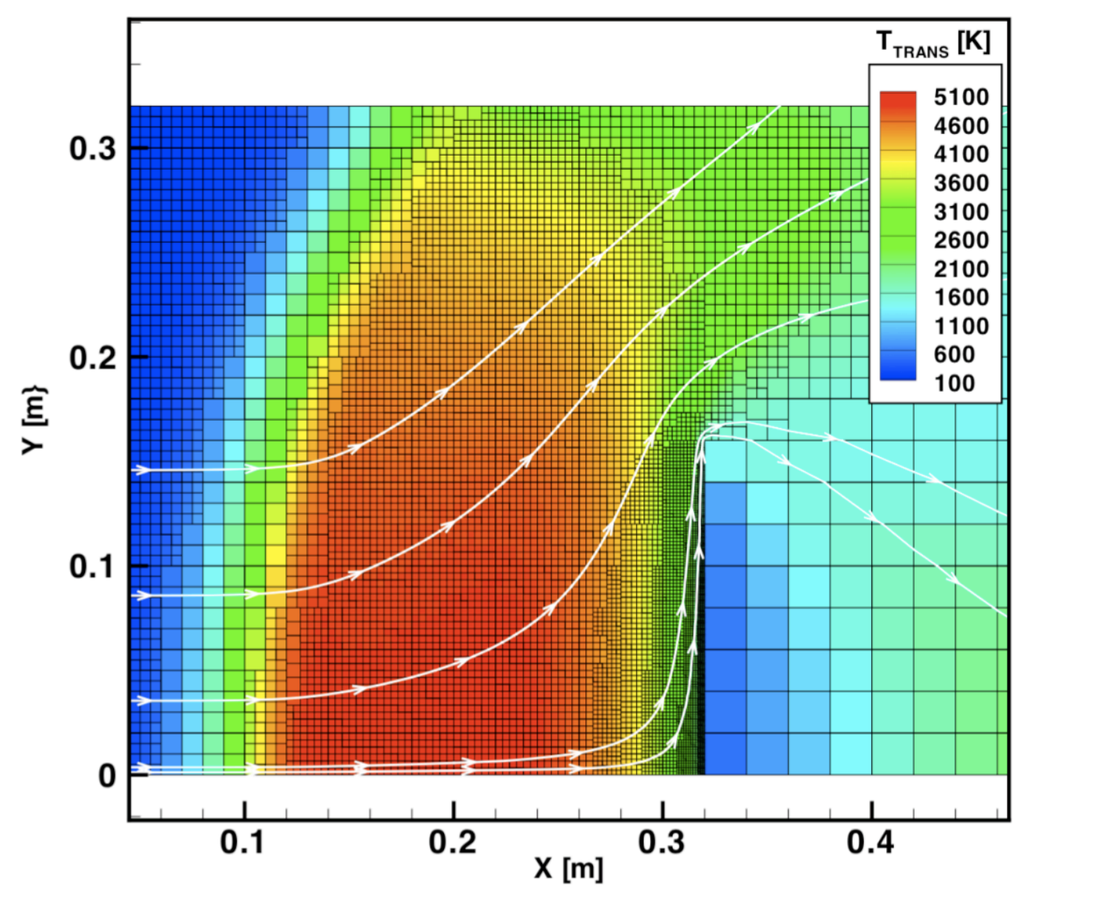
3 Adaptive Mesh Refinement (AMR) and Local Time Steps

An ideal DSMC simulation would precisely adapt cells to the local value of λ and would move all particles according to the local value of τ_c. This results in an accurate solution using the least number of computational cells and therefore particles. Each MGDS simulation begins with a uniform mesh and time step (image below). An initial solution is obtained and the cell size and time step are then refined based on the current λ-field.



Uniform mesh for rarefied Mach 15 flow of N_2 over a vertical flat plate.

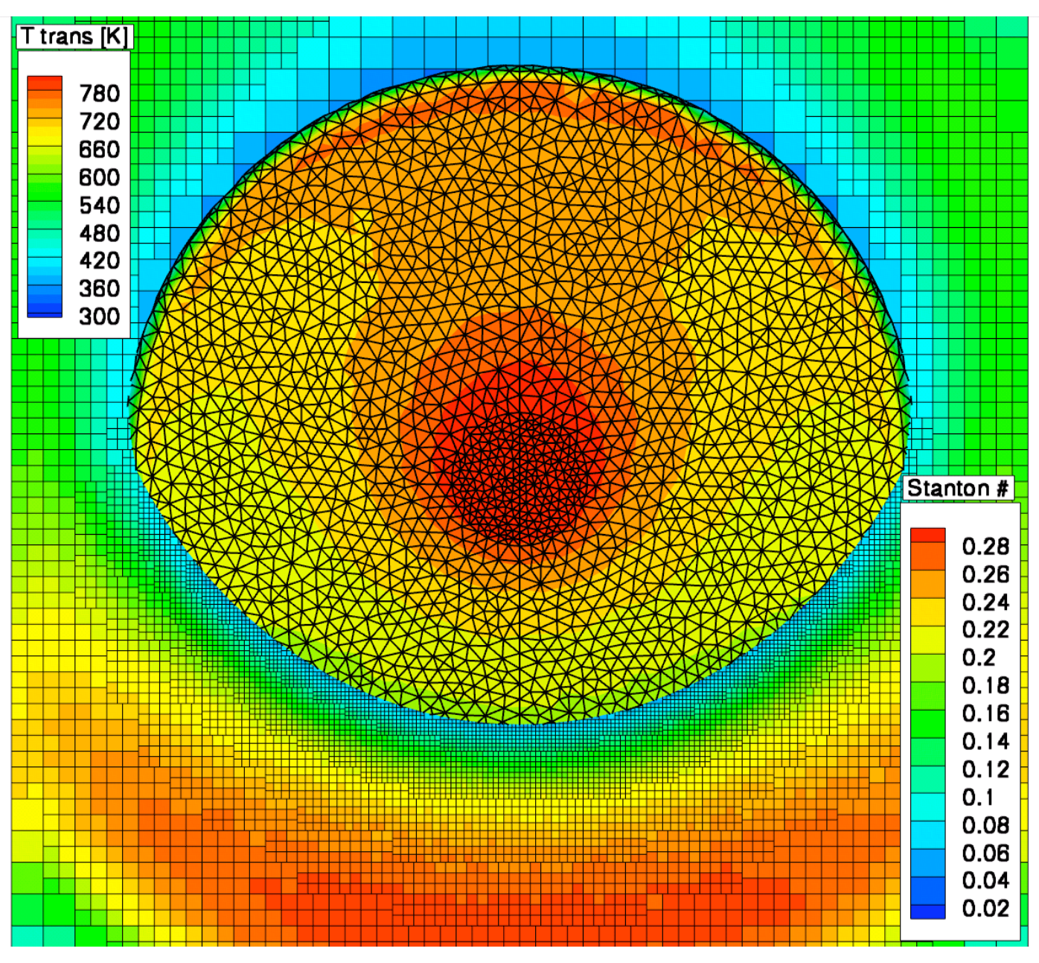
All particles are then re-indexed into the new mesh and the simulation smoothly continues. After only a few AMR procedures the mesh and local time step are precisely matched to the local values of λ and τ_c (image below). For this problem, the cell volume in the stagnation region is 30^3 times smaller than in the wake region which demonstrates the need for AMR, especially for 3D simulations.



Final refined mesh for rarefied Mach 15 flow of N_2 over a vertical flat plate.

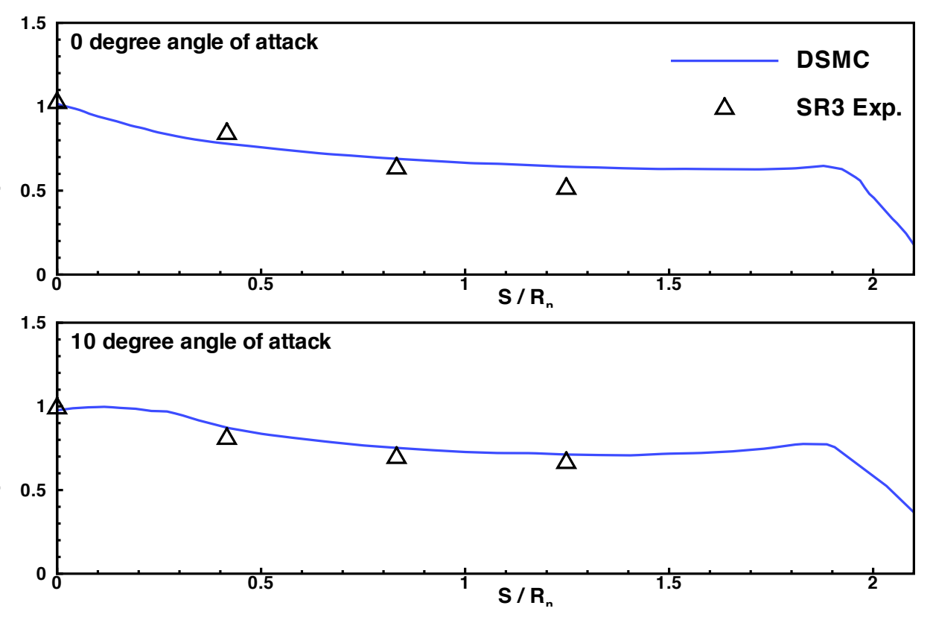
4 Surface Properties and Gas-Surface Interactions

Typical boundary conditions employed in DSMC simulations involve diffuse reflection and full thermal accommodation when particles collide with surfaces. Reflecting particles at random angles from a surface and setting their post-collision velocities in accordance with the local surface temperature reproduce standard no-slip and constant wall temperature boundary conditions under continuum conditions.



Flowfield mesh, surface mesh, and heating rate prediction at 10 degree angle of attack.

However, for non-equilibrium conditions involving a finite number of gas-phase and gas-surface collisions, a DSMC solution naturally predicts both velocity slip and temperature jump phenomena. It is also important to note that surface pressure, shear stress, and heat flux are directly computed by the change in momentum and energy of particles during surface collisions and not by evaluating gradients of velocity and temperature. Significant potential exists to incorporate molecular-based gas-surface boundary conditions within DSMC simulations.



Heat transfer coefficient along the symmetry line of the windward forebody. Conditions match SR3 tunnel experiments, Case 1.

5 Acknowledgments

This work was supported in part by the University of Minnesota Supercomputing Institute (MSI) and by the United States Air Force Office of Scientific Research (AFOSR).